**Phase-1 Submission Template**

**Student Name:** M.RANJITH

**Register Number:** 620123106092

**Institution:** AVS ENGINEERING COLLEGE **Department:** ELECTRONICS AND COMMUNICATION ENGINEERING

**Date of Submission:** 30.04.2025

# Problem Statement

With the exponential growth of digital media content, users often face difficulty in discovering movies that match their unique tastes and preferences. Traditional recommendation systems primarily rely on basic filtering techniques, which often fail to capture the complex and dynamic nature of user behavior. This leads to generic suggestions, reduced user satisfaction, and engagement. There is a pressing need for a more intelligent, adaptive, and personalized approach to movie recommendation that can understand individual user preferences, context, and sentiment in real-time. This project aims to solve this problem by developing an AI-driven matchmaking system that delivers highly personalized movie recommendations using advanced machine learning and natural language processing techniques.

# Objectives of the Projects

1. To build a system that recommends movies to users based on their interests.
2. To use AI to understand what types of movies each user likes.
3. To suggest movies by learning from user ratings, watch history, and preferences.
4. To improve suggestions over time as the system learns more about the user.
5. To make the recommendation system easy to use and helpful for all users.

# Scope of the Project

[

1. The system will recommend movies based on user preferences and past behavior.
2. It will use AI and machine learning to improve the accuracy of suggestions.
3. Users can give feedback, and the system will learn from it to give better recommendations.
4. The system will include a simple and user-friendly interface.
5. It can be used by streaming platforms, movie apps, or websites.

.]

# Data Sources

[

1. **MovieLens Dataset** – Provides user ratings and information about thousands of movies.
2. **IMDb Dataset** – Offers detailed information about movies, including titles, genres, ratings, and actors.
3. **Kaggle Movies Dataset** – Contains movie metadata, user ratings, keywords, and other useful details.
4. **TMDb (The Movie Database) API** – Gives access to movie details like cast, reviews, trailers, and posters.

# High-Level Methodology

*1. Data Collection:*

* + *Gather movie data (titles, genres, ratings) and user data (watch history, preferences).*

*2. Data Preprocessing:*

* + *Clean and organize the data to remove duplicates, fill missing values, and format it for analysis.*

*3. User Profiling:*

* + *Create a profile for each user based on their past movie choices, ratings, and behavior.*

*4. Recommendation Model:*

* + *Use machine learning (like collaborative filtering or deep learning) to match users with suitable movies.*

*5. Sentiment Analysis (Optional):*

* + *Analyze user reviews to better understand movie opinions and improve recommendations.*

*6. System Development:*

* + *Build a user interface that shows personalized movie suggestions.*

*7. Evaluation:*

* + *Test the system using accuracy metrics (like precision and recall) to see how well it recommends movies.*

# Tools and Technologies

**Programming Language** :

Python

**Notebook/IDE** :

Google colab

**Libraries** :

* + Pandas, NumPy – for data analysis.
  + Scikit-learn, TensorFlow, or PyTorch – for building AI models.
  + NLTK or spaCy – for sentiment analysis (if used).
  + Surprise or LightFM – for building recommendation systems.

**Optional Tools for Deployment** :

**Streamlit or gradio**

# Team Members and roles

**Ranjith .M – team lead**

**Oversees the project , performance data analysis,build and evaluate ml model**

**Prem kumar.S- data engineer**

**Collect and preprocess data ,integrates APIs,manage storage and data pipelines**

**Tharun.K – full stack developer**

**Develops and deploys the application (backend + frontend ), integrates ML models**

**Moulidharan .E -visualization &Q Aspecialist**

**Creates dashboard ,report and visual insights; test model accuracy and app performance**